# Consciousness for Robot Controller: Autonomous Mobile Robot Adaptation

Nelson Biedma[1],[1], David Trejo[1], Daniela López De Luise[1], Lucas Rancez[1], Gabriel Barrera[1], and Leonardo Isoba[1],

[1] AIGRoup, Engineering Department, Mario Bravo 1050,
C 1175ABT Buenos Aires, Argentina
info@tbrain.com.ar

**Abstract.** This paper focuses on the expansion of a conscious framework named ALGOC, and the benefits that derive from using such models as part of a mobile robot controller. There is a first description to the mobile robot problem and the relevance of an optimized intelligent controller. The architectural main features are described. Some test cases with and without a conscious controller show that there is an improvement of the robot´s behavior performing an optimized source-to-target path.

**Keywords:** Autonomous Mobile Robots, Consciousness, Codelets.

## 1. Introduction

This paper makes a small statement of the Autonomous mobile problem, its relevance and how it can be addressed by using a model called ALGOC, which constitutes the core of the FIC robot prototype. It defines the essential features of a Concept Model, and how real-world objects are modeled and logically handled by a conscious based system. The rest of this paper is organized as follows: there is an introduction to the mobile robot problem (section 2), a description of the FIC prototype design, architecture and hardware (sections 3, 4 and 5 respectively), some representative test cases (section 6) and the preliminary results (section 7). Finally, there are some conclusions and a proposal of future work to be done (section 8).

## 2. Autonomous mobile robots

Current mobile robot development technology is mostly focused on robots capable of moving based on predefined ad hoc routines. These approaches are usually based on the knowledge or experience acquired by the developers according to the robot's behavior observation [1]. Also, they are defined, or at least tuned, for a certain environment, in order to obtain a better response that takes into account the

[1]Author is for project FIC in AIGroup at Palermo University.

particularities of a given context [2]. The global life cycle of the robot can be summarized with the following steps:

- Develop a strategy to rule the behavior of the robot.
- Perform basic steps of the strategy.
- Sense current environment.
- Detect non-expected, expected but inaccurate, and missing behaviors.

This developing model may provide an accurate, expected behavior after looping several times on this life cycle. Nevertheless, it involves limitations that are inherent to the model:

- Accurate behavior implies limiting the capabilities to a specific, particular environment.
- Appropriate behavior will occur only on foreseeable situations at developing time.
- Behavior learning relies on human involvement, since behavior corrections to be done require human survey and corrected behavior implementation requires reentering data or algorithm amendment, making behavior evolution static and manual.
- Development can hardly be reused on a substantially different scenario.

The goal of FIC is to achieve an autonomous mobile robot [9] based on a new generation behavioral paradigm that is not a next-step improvement over the current one, but a new re-engineered system built from the ground up, characterized by computational intelligence, a cognitive robotic system [6], consciousness, context autonomous learning and awareness qualities. This new approach can deal with a standard robot life cycle and also overcome limitations mentioned previously.

## 3. FIC global architecture

Neither FIC is based on preset procedural rules nor uses heuristic guidelines but it implements automated concept learning and inferring. The behavioral paradigm, based on an initial knowledge that should be detailed and complete (based on developed algorithms, input world mapping and sensing data) is replaced by one in which the robot is released into the environment with very simple world knowledge, but provided with the ability to dynamically perceive, learn, remember, associate, and modify its reactions, behavior, and problem solving strategies according to experience [5], [8]. As a consequence, the robot can also adapt to an unknown and changing context. These kinds of context resemble very much the real world. Obstacle objects are built from perception. These are known as similar or equal, through common attribute recognition, such as shape, size or location. With current information and relevant past knowledge, the robot can take the appropriate decision.

This leads to a complex non-deterministic model that intends to resemble the learning process. In this way, the robot´s behavior should mature according to its experience automatically at run time. All the process is performed without code recompiling or other input data than that sensed autonomously.

Raw environmental data is sent to a smart controller, and converted into percept objects (by ALGOC model concept conversions) [7], which are eventually conceptualized as real world objects, such as obstacles (including moving obstacles) or desired arrival points. After recognition of the obstacle objects, and automatic evaluation of current localization [10], [11], [12], the smart controller evaluates strategies and sends the best one to the robot´s real-time controller. It receives the advices and commands and has two alternatives: ignoring or taking them according to robot´s current priorities. Fig. 1 shows this feedback system. RTA (Robot Task Adviser) is the intelligent controller that provides middle and long-term strategies. The RTC is the real-time controller in FIC [3].

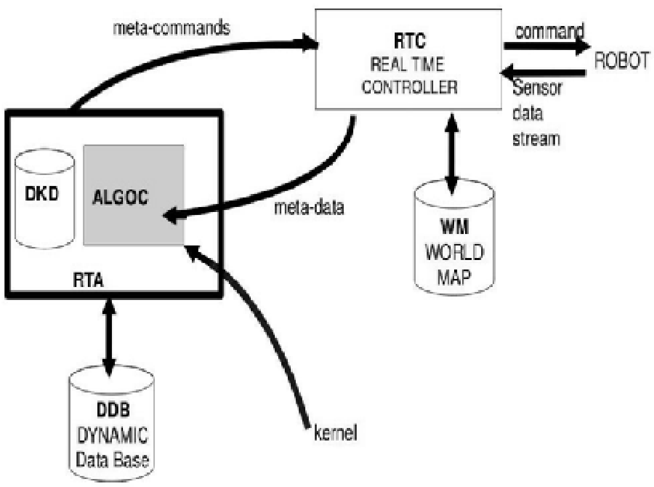


**Fig. 1.** FIC architecture.

The RTC (Real Time Controller) runs and forwards the robot basic immediate decisions. This controller is very simple compared to RTA, providing a response in a shorter time. It has a higher priority command execution under situations that require rapid response (for example danger situations).

The described dual feedback system (RTA / RTC) provides two different behavior criteria. The first one grants priority to achieving a smart strategy, and the other one to a fast processing, retrieval and execution for current real time requirements.

## 4. FIC functionalities

FIC is expected to provide adaptive responses that will be increasingly sharp and appropriate for a specific goal and environment, based on previous experience, attempts, and different degrees of success and failure. Hence, each subsequent path and speed combination becomes closer to optimal.

At the current development stage, this autonomous mobile robot provides a good response in static indoor flat environments. Non-flat and non-smooth surfaces are outside the current FIC development scope, along with inclined surfaces, even if flat and smooth.

The current behavior is derived from the ALGOC general framework, which is a model implemented to build systems able to learn and adapt by the construction of concepts [3].

The model is good for applications ranging from scientific, technological, up to industrial usages [4].

## 5. Hardware Description

The selected hardware platform has a main programmable module in Java language. It has also three servomotors and several mobile pieces.

Robot has two sensors that allow it to analyze the environment, a push-button and an ultrasonic sensor. The push-button is located in the front of the robot and it is used for detecting obstacles that ultra sonic sensor cannot detect. The main goal of the ultrasonic sensor is to detect obstacles and feed back the distance between these and the robot at any time. Data is sent from the robot to an intelligent controller that makes a new concept using it, and generates a new world map.

Hardware has an ultra sonic sensor with a scanning Frequency of 40 kHz. The average wheel speed is 21.05 cm / sec.

## 6. Test cases

Robot controllers were tested in a room of 200 cm x 200 cm. Three tests cases were performed: with one, two and three obstacles, having a unique target in all cases. The initial point and target were identical in all cases. Obstacles were rectangular, sized 20 cm x 60 cm.

In all cases the robot was tested with and without RTA. In the first alternative the paths were performed in segments, while in the later the paths were smooth curves. At each joint of two segments there is a delay between stop and restart of 1.16 seconds. The maximum speed was 21.05 cm/seconds.

In these tests, the floor was smooth and even and had neither slopes nor steps. Test cases are detailed and analyzed below.

### 6.1. Test case 1: simplest world map

The first test was made in a rectangular room with only one obstacle in front of the initial path of the robot (see Fig. 2).

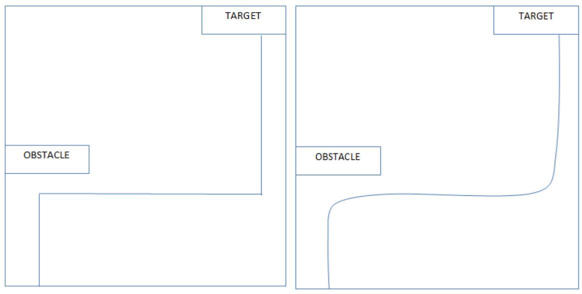


**Fig. 2**. Simple controller (left) versus ALGOC (right) with one obstacle.

The test with RTC is showed on the left half of the figure. The path performed had 364 cm length, and the total time spent was 19 seconds at a speed of 21.05 cm/sec and two delays of 1.16 seconds each.

The same test was repeated with RTA. The total length of the trace was 274 cm walked in 13.016 seconds.

### 6.2. Test case 2

The second test was made on the same rectangular room as the first, and had two identical obstacles deployed in two opposite sides of the test room (see Fig. 3).

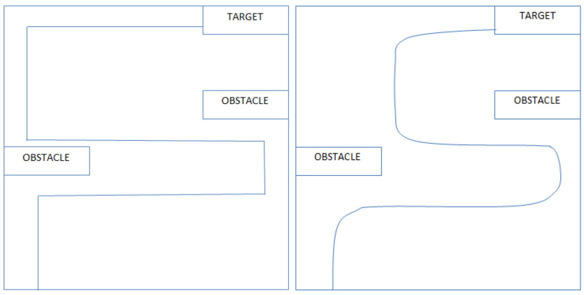


**Fig. 3.** Simple controller (left) versus ALGOC (right) with two obstacles.

It is so to observe the behavior of the controller when finding a new obstacle after avoiding the first one.

When the RTC controller begins to run in this context, the length of the path is 655 cm split in six segments with a delay of 1.16 seconds in every junction of two segments, with a total delay of 5,8 seconds. The total time spent was 36.916 seconds.

The same test, run with, RTA, showed a path length of 475 cm, walked in 22.563 seconds.

### 6.3. Test case 3

The third test was made in the same rectangular room as the preceding ones, with three obstacles: those on test case 2, plus a third one that partially obstructs the free space between the first two (see Fig. 4).

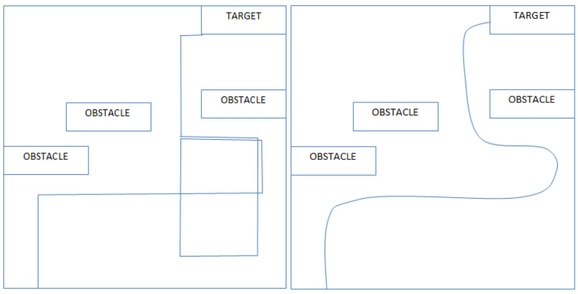


**Fig. 4**. Simple controller (left) versus ALGOC (right) with three obstacles.

It is so to observe the behavior of the controller when finding subsequent obstacles. When the RTC controller begins to run in this context, the length of the path is 731 cm split in ten segments with a delay of 1.16 seconds in every junction of two segments, with a total delay of 10.44 seconds. The total time spent was 45.166 seconds.

The same test, run with, RTA, showed a path length of 376 cm, walked in 17.867 seconds.

## 7. Conclusion and Future work

The controllers RTA and RTC were tested under similar conditions, with one, two and three obstacles. The observed behaviors were remarkably better for the RTA controller, with fewer seconds spent to perform the traveling from source to target. Three tests demonstrate also more efficiency regarding the path length. This suggests that RTA would be a better solution to the autonomous mobile robots under these conditions.

For future work, it is pending to test the robot´s behavior with mobile obstacles. It is also interesting to extend the concept model to adapt to ground sensors.

## References

1. Sillitoe, I., Elomaa, T.: Learning Decision Trees for Mapping the Local Environment in Mobile Robot Navigation.

2. Kavraki, L., Svestka, P., Latombe, J., Overmars, M.: Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. IEEE International Conference on Robotics and Automation (1996)
3. López De Luise, D., Barrera G., Franklin S.: Robot Localization using Consciousness. Journal of Pattern Recognition Research (2011)
4. Vanderborght, B. Sugar, T. Lefeber, D.: Adaptable compliance or variable stiffness for robotic applications. In: Robotics & Automation Magazine, IEEE, pp. 8--9. IEEE Press, (2008)
5. H. F. Ebel, C. Bliefert, and W. E. Russey, The Art of Scientific Writing : From Student Reports to Professional Publications in Chemistry and Related Fields. Weinheim: Wiley-VCH Verlag GmbH & Co. KGaA, 2004
6. K. A. Ericsson, and W. Kintsch, Long-term working memory. Psychological Review, 102, 211-245., 1995
7. J. L. Posadas, J. L. Poza, J. E. Sim, G. Benet, and F. Blanes, Agent-based distributed architecture for mobile robot control. Engineering Applications of Artificial Intelligence, 21(6), 805-823., 2008
8. H. P. Nii, The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. The AI Magazine, Summer 1986, 3853., 1986
9. S. Franklin, and A. Graesser, Is it an Agent, or just a program?: A Taxonomy for Autonomous Agents. Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag., 1996
10. C. Eberst, M.Andersson, and H.I. Christensen, Vision-based door-traversal for autonomous mobile robots. Inst. for Real-Time Comput.-Syst., Tech. Univ. Munchen; Proc. 2000 IEEE/ RSJ International Conference on Intelligent Robots and Systems., 2000
11. D. Lecking, O.Wulf, and B. Wagner, Localization in a wide range of industrial environments using relative 3D ceiling features. Emerging Technologies and Factory Automation, IEEE International Conference on ETFA 2008. pp 333 - 337, 2008
12. S. Kolski, D. ferguson, M. Bellino, and R. Siegwart, Autonomous Driving in Structured and Unstructured Environments. IEEE Intelligent Vehicles Symposium, 2006